

Python: module browser.gui_extend_menus

browser.gui_extend_menus

[index](#)

```
# The PCMDI Data Browser Extend Toplevel Menus - gui_extend_menus module
#
#####
#
# Module:          gui_extend_menus module
#
# Copyright:       "See file Legal.htm for copyright information."
#
# Authors:        PCMDI Software Team
#                  Lawrence Livermore National Laboratory:
#                  support@pcmdi.llnl.gov
#
# Description:    PCMDI Software System browser Extend Toplevel Menus.
#
# Version:        4.0
#
#####
#
#-----
# NOTE: need to use version of Python that imports Tkinter and Pmw
#-----
```

Modules

<u>Tkinter</u>	<u>browser.gui defined variables</u>	<u>os</u>	<u>types</u>
<u>main</u>	<u>browser.gui message</u>	<u>string</u>	
<u>gui_support.gui_color</u>	<u>browser.gui saved settings</u>	<u>sys</u>	
<u>browser.gui control</u>	<u>browser.gui select variable</u>	<u>tkFileDialog</u>	

Classes

create

```
class create
```

```
#-----
#
# Start of Popup Dialog
#
#-----
# Extend Toplevel Menus Dialog Popup
```

```
#-----
```

Methods defined here:

```
__init__(self, parent)
```

```
add_function(self, parent, evt=None)
```

```
#-----  
# Add a function to a pre-existing menu from the main menu bar  
#-----
```

```
add_top_menu(self, parent, evt=None)
```

```
#-----  
# Add a new menu to the main menu bar  
#-----
```

```
browse_files(self, parent)
```

```
#-----  
# Open a tkinterFileDialog window and get filename from user  
#-----
```

```
change_description(self, parent, evt=None)
```

```
#-----  
# Change menu description  
#-----
```

```
clear_top(self, parent)
```

```
#-----  
# Clear "Menu name" and "Description" from the "Create user menu"  
#-----
```

```
del_top_menu(self, parent)
```

```
#-----  
# Remove a menu from the main menu bar  
#-----
```

```
delete_function(self, parent)
```

```
#-----  
# Delete a function from the main menu bar  
#-----
```

```
do_nothing(self, result)
```

```
#-----  
# Use this function to "lock" a field (bind <KeyPress> to do_nothing)  
#-----
```

```
file_or_import(self, parent, result)
```

```
#-----  
# Toggle between file and import mode. Change the appearance of  
# group2 and group2a appropriately  
#-----
```

```

get_functions(self, result=None)
    #-----
    # Get functions from file or import and populate the function
    # pull down menu
    #-----

import_functions(self)
    #-----
    # Get functions from an import
    #-----

populate_group2(self, result)
    #-----
    # When user selects a menu name from the "Modify user menu" w
    # populate all appropriate fields with current menu informat
    #-----

rename(self, evt=None)
    #-----
    # Rename a function
    #-----

select_function(self, parent)
    #-----
    # Change the value of "Function name" field to value of "Func
    #-----

```

Functions

```

find_modules()
    #-----
    # Return the Modules that have been imported
    #-----


find_py_files(parent)
    #-----
    # Find the py files in the directory
    #-----

restore_menus(parent)
    #-----
    # Restore Menus in the Main Menu Bar
    #-----

returned_functions_and_instances(self)
    #-----
    # Return the Module's function list
    #-----

wrap_balloon_help(contents, max_width=None)

```



```
# Make formatting of help balloons easier
```

Data

```
Pmw = <Pmw.Pmw_1_2.lib.PmwLoader.PmwLoader instance>  
fn = '/pcmdi/halliday1/PCMDI_GRAPHICS'  
l_menu = []
```